

# ;login:

THE UNIX NEWSLETTER

VOLUME 2 NUMBER 9

OCTOBER 1977

Page 1

## NOTICE

This document may contain information covered by one or more licenses, copyrights, and non-disclosure agreements. Circulation of this document is restricted to holders of a license for the UNIX or Mini-UNIX software system from Western Electric. Such license holders may reproduce this document for uses in conformity with the Unix license. All other circulation or reproduction is prohibited.

## RUNNING LATE

We are still playing "catch-up". This newsletter is being prepared in mid-December and will be mailed before Christmas. The contents reflect the true calendar date, not the nominal month on the mast-head.

The next issue, nominally November 1977, will be mailed early in 1978 and will complete this volume. Enclosed with that issue will be an invoice for Volume 3 (1978).

## SOFTWARE DISTRIBUTION

We are well along on the backlog of tape orders. Almost all of the old orders which included a user tape have been mailed. A temporary shortage of new tapes due to vendor change has ended, and we should have eliminated the entire backlog by the end of January 1978.

As of now, there has not been sufficient new material submitted to justify a fourth software distribution.

## NEW TORONTO RELEASE

The Toronto release has been recast into standard distribution format and is available as announced in the last issue. You are reminded that a new Toronto license must be submitted even if you had a previous license. The form is in the last issue.

## MEETINGS

The minutes of the fall West Coast Meeting are included in this issue.

Your attention is called to the announcement, in this issue, of a one day West Coast Meeting in January.

## USERS GROUP

The membership in the Users' Group now exceeds 250. All of the work on the newsletter, the software distribution, the mailing and invoicing is done by contributed time of volunteers. We have reached a size where the answering of telephone inquiries about either subscriptions or software severely impacts our ability to do our normal work. We must request that all inquiries about the above items be made in writing to the appropriate address in the box below.

## MAILING LIST

Having discussed the issue of the proper use of the mailing list with several concerned people, we would propose the following "rules". The mailing list is the property of the newsletter and is provided to you for use by your installation in order to encourage the interchange of information. The giving of this list to vendors is not in accord with our providing you with the list.

Since all of those we spoke with felt they did not object to the receiving mailings from vendors but did object to vendors being given their addresses and telephone numbers, we have established a policy of doing such mailings to the membership for vendors for a nominal charge. If you are approached by vendors, please refer them to Armand Gazes at the address below for details.

Your comments upon and objections to this policy are solicited (in writing).

Address editorial material and software submission to

Melvin Ferentz  
c/o CUNY/UCC  
555 West 57 Street  
New York, N.Y. 10019

Subscription requests, payments and address changes should be addressed to

Armand Gazes  
Physics Department  
Brooklyn College  
Brooklyn, N.Y. 11210

The University of Wollongong  
P.O. Box 1144  
Wollongong, N.S.W., AUSTRALIA

Ref. JR:gmg  
Telex AA 29022

21st November, 1977

Professor Melvin Ferentz  
Brooklyn College of CUNY  
Brooklyn, New York 11210  
U.S.A.

Dear Professor Ferentz,

Could you please enter our subscription to UNIX News. Our Interdata 7/32 UNIX has been running in production since July hence we were quite surprised to read Ken Thompson's letter in the last UNIX News concerning the Bell Laboratories version of UNIX for the Interdata 8/32 machine.

Since our version of Interdata UNIX suffered very little from memory requirement increases or performance degradation it seems that it established quite clearly the portability of UNIX via a "C-machine".

Yours sincerely,

R. Miller

J. Reinfelds.

UNIVERSITY OF CALIFORNIA, SANTA BARBARA

BLAKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO



SANTA BARBARA • SANTA CRUZ

COMPUTER CENTER

SANTA BARBARA, CALIFORNIA 93106

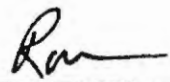
November 15, 1977

Mel Ferentz  
c/o CUNY/UCC  
555 West 57 Street  
New York, New York 10019

Dear Mel:

West Coast Unix users will have a one-day meeting at the Santa Barbara Campus of the University of California on January 20, 1978. The meeting will begin at 8:30 a.m. in the Program Lounge of the University Center. We hope that most Unix folks on the west coast will be able to attend.

Sincerely,

  
\_\_\_\_\_  
Ron Jeffries  
Programmer, User Services  
Computer Center

RJ:bd

DEPARTMENT OF COMPUTING  
AND INFORMATION SCIENCE

Queens University  
Kingston, Canada  
K7L 3N6

October 6, 1977

Dr. Mel Ferentz  
Physics Department  
Brooklyn College  
Brooklyn, N.Y.  
U.S.A.

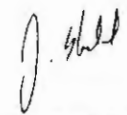
Dear Dr. Ferentz:

We currently have preliminary drivers for DZ-11 and RK-06. We would be glad to release them to individual sites, but do not want to release them for distribution as they will be vastly modified soon.

The DZ driver does not support modem control.

The RK-06 driver requires disk packs with no bad sectors. (This will be fixed in a subsequent version).

Sincerely,

  
J. H. Kulick  
Assistant Professor

JHK:ap



**BASSER DEPARTMENT OF COMPUTER SCIENCE**

School of Physics (Building A28),  
University of Sydney, N.S.W. 2006

10 November 1977

Professor Melvin Perentz,  
C/- CUNY/UCC,  
555 West 57 Street,  
NEW YORK, N.Y. 10019  
UNITED STATES OF AMERICA

Dear Professor Perentz,

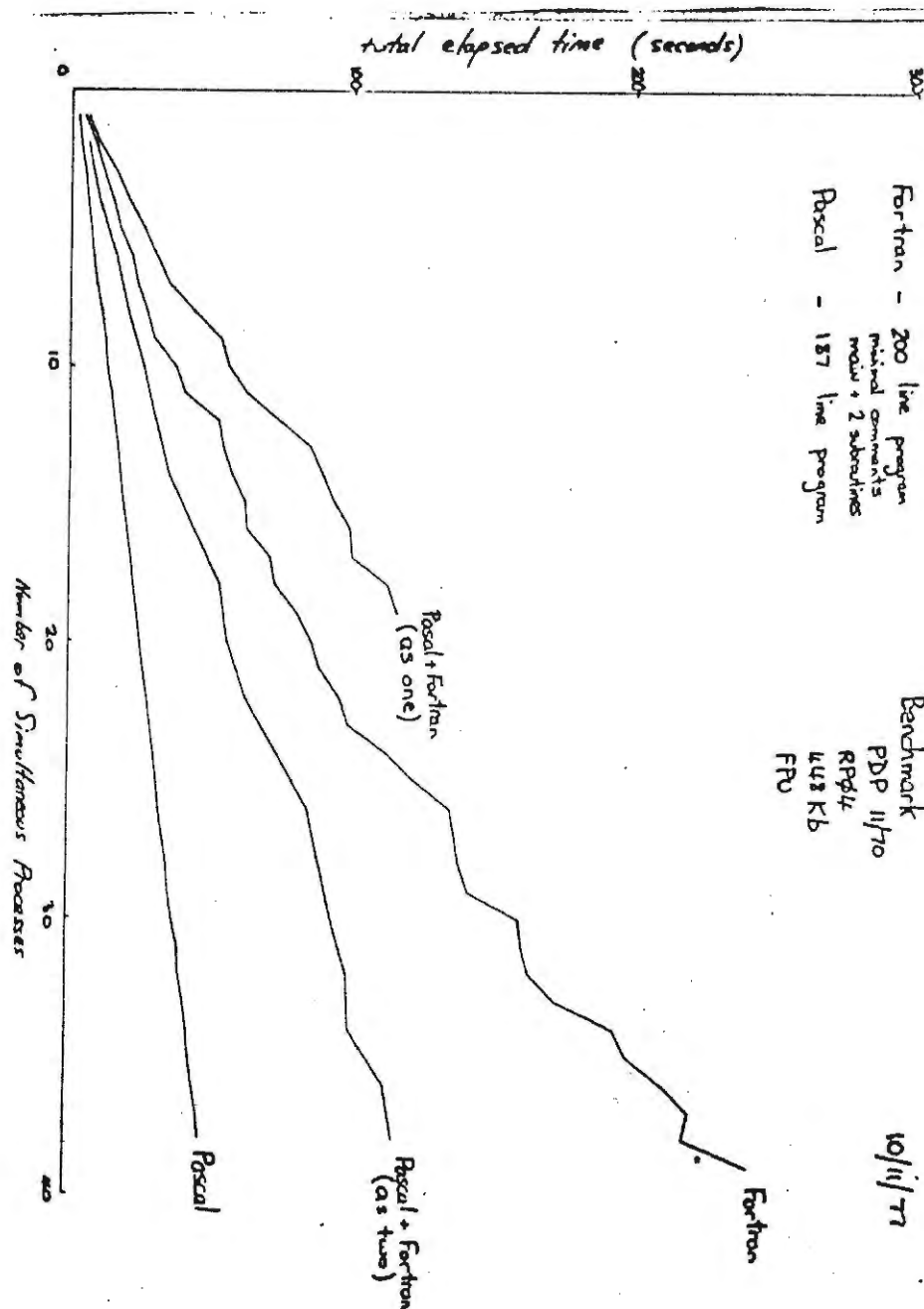
At Sydney University UNIX has been running on an 11/40 with great success for some time and winning over 'all'. A larger machine to service the teaching/research needs of our Department is now under serious consideration. As a consequence we have been conducting a number of tests on an 11/70 at the University of New South Wales to determine UNIX's ability to handle a large number of student terminals. We hope to convince ... that an 11/70 + UNIX will meet our needs in the best possible way.

During these tests we have come across a number of apparent bugs/glitches in UNIX when it is under a heavy load, the main areas affected being file system integrity and scheduling (process hanging). As we really cannot delve deeply into these problems at the present time. I would greatly appreciate it if you could air mail a copy of the "diff listing of changes/bug fixes to UNIX" to me as soon as possible. This is probably vital to our case for UNIX.

For your information (and possibly others) I include a chart of our results to date (a late night session last night!).

Yours sincerely,

(Piers Lauder)





Psychologisch Laboratorium

# KATHOLIEKE UNIVERSITEIT

FACULTEIT DER SOCIALE WETENSCHAPPEN

Prof. M. Verents  
Brooklyn College of CUNY  
Brooklyn NY 11210  
USA

NIJMEGEN, September 6, 1977  
no. 5700/GR/wh

Dear Professor Verents,

As you might have noticed, I have switched departments.

About 3 months ago, I called you for information about heavy loaded UNIX systems, and file shipping between UNIX and IBM 360/370 machines. You directed me to Lew Law at Harvard, and to Chuck Panner at Berkeley. From them, I got very valuable information for a follow-up on my initial proposal.

The proposal is still in the mill, and may never get out, but my new department is interested very much in the RJE/file shipping part of my proposal. Therefore, I have included a note to be posted in UNIX News. At the same time I wrote to Rick Haight at Bell Labs, a pointer that I didn't follow at the time I called you.

Thanks a lot.

Sincerely,

*George Rolf*  
George Rolf

RJE and file shipping software sought.

Who has RJE and/or file shipping software between UNIX and IBM OS-MVT/VS1/VS2/MVS systems.  
A 2780 emulator for UNIX would be great to start with.

Please contact:

George Rolf  
Psychological Laboratory  
Erasmuslaan 16  
Nijmegen  
The Netherlands

Bijlage:

ERASMUSLAAN 16 NIJMEGEN — TELEFOON (050) 51.2633



# QUEEN MARY COLLEGE

UNIVERSITY OF LONDON

## DEPARTMENT OF COMPUTER SCIENCE AND STATISTICS

Head of Department: Professor DE Barton  
Professor of Computing Science: IM Khabaza  
Professor of Theoretical Computer Science: PJ Landa

MILE END ROAD

LONDON E1 4NS

Tel. 01-880 4811

20th October 1977

Professor Melvin Ferents  
c/o CUNY/UCC  
555 West 57 Street  
New York, N.Y. 10019

Dear Professor Ferents,

We have been successfully running UNIX on a 11/40 for over two and a half years and recently increased the number of mounted filestores on our system to seven. Since that time we regularly experienced the following: the processor would enter the 'idle' routine, service interrupts but not find any runnable processes to execute. Terminals at shell level were able to run shell commands, e.g. `chdir` without an argument, wait, etc. but would hang up if any disc access was required.

We traced this fault to a deadlock over the block buffers (our system is considerably larger than the one distributed and we only have room for fifteen of these buffers). We originally suspected that deadlock was occurring because of the fact that four processes could be in 'exec' simultaneously, however, reducing `MAXEC` to two did not cure the problem. Further investigation revealed that deadlock is possible whenever

`NBUF` - no. of mounted filestores < no. of processes  
executing 'bread' simultaneously

This is theoretically possible irrespective of the number of buffers and the number of mounted filestores (on the assumption that `NBUF` - no. of mounted filestores < `NPROCS`).



# QUEEN MARY COLLEGE

UNIVERSITY OF LONDON

## DEPARTMENT OF COMPUTER SCIENCE AND STATISTICS

Head of Department: Professor DE Barton  
Professor of Computing Science: IM Khabaza  
Professor of Theoretical Computer Science: PJ Landa

MILE END ROAD

LONDON E1 4NS

Tel. 01-880 4811

However, systems with a large number of buffers or a small number of mounted filestores will rarely (if ever) experience this fault.

This bug may be easily cured by not setting off a read-ahead if no free buffers are available. That is to say, the condition in 'bread' under which read-ahead is initiated:

```
if( rablno && !incore(dev, rablno) ) {
```

should be changed to:

```
if( rablno && !incore(dev, rablno)
    && (bfreelist.av_forw != &bfreelist) ){
```

yours faithfully,

Richard Bornat  
Jon Rowson  
Ben Salama

*installed*  
*4 Jan 78*  
*TEF*

# UNIX USERS' GROUP Third West Coast Meeting

The following is a rough account of the West Coast meeting of the UNIX Users' Group. It is divided into sections roughly corresponding to the major topics discussed at the meeting.

## 1) The West Coast UNIX Users

Two committees were established to oversee future action oriented toward the West Coast user community. The first is a site selection steering committee consisting of John Bass, Gerry Harksdale and Mike O'Brien. The second is a topics committee consisting of Mark Kampe and two other people (I forget who, unfortunately -- Ed.).

It was agreed to have roughly three meetings per year. The next meeting will be at the UC Santa Barbara campus, to be followed by a June meeting.

Since there is a fairly wide class of known bugs and tuning items associated with the system (the TU-16 driver as distributed does not work), it was suggested that a new users' Welcome Wagon be instituted, by having sessions at the meetings oriented to new users. In addition, as an aid to Unix system's programmers who are encountering difficulties with their Unix systems, certain experienced members of the community have volunteered their services for BRIEF question-answering. The volunteers for this service were:

|                  |                                      |
|------------------|--------------------------------------|
| John Bass        | (805) 544-4407 < 10am, BASS@SRI-KL   |
| Mark Kampe       | (213) 625-4733, MARK@UCLA-SECURITY   |
| Lauren Weinstein | (213) 825-4733, LAUREN@UCLA-SECURITY |
| Jeff Schriebean  | (415) 642-1035                       |

These people have varying degrees of reachability. Kampe and Schriebean can be quite hard to find.

While it was generally felt that there should be a centralized database of installation descriptions, no one could come up with an immediate scheme which would allow 1) a nicely formatted input document which could be kept in a database, and 2) a description other than a boringly detailed catalog of hardware configuration which does not contain a description of the type of work being done and software being produced, kernel changes made, etc. Jim Lieb from ISI has volunteered some cycles, some disk and some of his time to try to make such a database a reality.

## 2) Accounting

Rand's accounting scheme was described, in which separate user directories are charged to different account numbers under

the owner. Process accounting is handled by new entries in the U structure which are collected by an expanded "wait" system call. The collection of such data is done by "init" which manages the accounting file itself.

UC Berkeley described the "disk quota" system implemented by Ken Thompson during his stay there, wherein a new type of inode was created. A file in a directory named ".q" may be created by the super-user, whose address list does not contain file blocks, but rather a usage count and allowable limit on blocks in use in that directory and all sub-directories. Every time a block is allocated, the .q entries must be updated all the way up the tree. All .q files which are currently applicable to opened directories are kept around in the in-core inode table. This scheme is combined with one wherein users with group ID's greater than 127 cannot access files owned by any other user.

## 3) Text Processing

Rob Lawrence of SRI described the basic capabilities of the Version 7 Phototypesetting package in the context of the Editorial Processing Center work now being carried on at SRI under the direction of Dr. Oliver Whitby. He is interested in discovering whether a system can be built which could be used by people with little computer experience to process documents of a fairly high degree of complexity.

Gerry Harksdale of the Naval Postgraduate School in Monterey described the work being done there in producing documents using a wide variety of character sets on a Versatec printer/plotter. Some character sets are based on the Hershey Digitized Character Set while others have come from SAIL. Documents are produced by the following mechanism:

document -> troff -> virtual typesetting system -> versatec

font files

They have an editor for fonts to fix up characters that were made ugly by the translation from a vector list to a digitized matrix. So far there is no nice way to do variable-width fonts. Currently the system runs at 60% of the maximum hardware speed, using the Simultaneous Print/Plot mechanism of the Versatec. MPGS is willing to make its font files available through the UNIX Users' Group Software Distribution Center.

## 4) Multiprocessing

The Naval Postgraduate School is also doing research on a multiprocessor version of UNIX. They have two 11/50's communicating through shared memory, and an 11/34 with a Vector General Display also communicating with shared memory. Their UNIX divided the text, data, and bss sections of a program into three different maps. The system runs at about the same speed as regular

UNIX, with somewhat greater fragmentation of the freelist. They also have a shared-segment UNIX which permits allocation of common core to different processes. The entire scheme is designed to allow the allocation of data segments to particular core areas which are shared with a separate display processor. Swapping still sends out all three portions of a program. Swapping policies have not been investigated.

The application of this system is to a signal processing and display system, which is controlled by a fourth, special-purpose processor also communicating through shared memory. It has a 16-channel A/D converter and an array processor to handle signal data. The data can either be read by one 11/50 for use in data reduction, or by the other 11/50 for formatting and transmission to the 11/34 for real-time display.

#### 5) Reliability of Files

A rather confusing discussion on file system reliability resulted in the conclusions that UNIX needs a stand-alone restoration program, since neither ROLLIN nor PRESERVE save the spare sectors on a disk. Version 7 has a fair variety of file system improvements. There is a current error worth looking at: if the section of the system which updates inodes on the disk gets a read error on the block containing the inode it wishes to update, it will not check for the error. Instead it updates the inode in the bad block and writes the bad block out. This can cause groups of 16 bad inodes all right together, behavior which has been observed in the past.

Berkeley has modified "dump" and "restor" to use a rather larger record size on output, which saves tape.

UCLA has a utility for dumping the images of multiple disks to a single tape. This requires a multi-file tape driver which UCLA and other installations have. They also have a utility, not unlike *tp*, which has the ability to append files to the end of a tape.

There was a brief discussion of bad blocks, during which it was mentioned that version 7 will collect bad blocks into a special inode (#2) on each file system.

Both John Bass of SKI and the Rand UNIX people will be working on a more general method of archiving and backup in the coming six months or so, to address the problem of saving space and managing the extremely large disk drives now becoming available. It will combine forced and requested archiving of old files and their deletion and restoration to the file system. It is hoped that these mechanisms will have a very simple user interface (wherein the user is not responsible for allocating or managing tapes), and that they will be suitable both for use by individual users and for system managers.

#### 6) Secure UNIX

Mark Kampe of UCLA described the work currently being done on a "data-secure" unix system being developed by using program-verification techniques. Since these techniques are still in their infancy, the relevant portions of code must be small. Hence, their system runs a very small kernel, with the scheduler and the file handler running in supervisor and user spaces, and communicating with the kernel through a restricted set of system calls. Security policy is not part of the kernel; it handles only enforcement. For the sake of verifiability, the kernel is non-interruptible. Super-user-restricted calls have for the most part been replaced by more innocuous calls implementing the actual functions desired, or thrown away entirely in favor of a different protection scheme.

Currently the work has verified substantial portions of the kernel, and a prototype file/policy manager and scheduler have been written. The prototype file/policy manager can support a broad range of policies, but currently only a type of military security rule is enforced. An ARPA Network control program to handle multiple levels of secure traffic has been designed and implemented. The work on the operational prototype is continuing, including extensions in functionality and performance optimization.

Ford Aerospace and TKW are working on secure UNIX projects, attempting to transfer the technology developed at MITRE and UCLA into a production Data Secure Unix system. It is hoped that the resulting system will be available to non-DoD sites also (the UCLA work will be in the public domain). Design and development should be done within two years. The expected target is the government installation desiring an off-the-shelf, supported, small, secure system.

#### 7) Touch-tone UNIX

Lauren Weinstein of UCLA has, in the last two months, given UNIX the ability to communicate over a touch-tone phone in full ASCII. A user with a touch tone phone can do anything that he could do with a terminal (send and receive mail, edit and compile programs, play games, etc.) A keystroke code has been devised which allows full ASCII to be generated with fairly intuitive combinations of button presses. A VOTRAX synthesizer speaks the output over the phone using a substantially enhanced version of the Bell Labs "speak" program (a rule directed, english text to phoneme translator). A device driver was written to interface to the votrax and a touch-tone modem, and a program was written to take the touch tone input, translate it into ASCII and feed it into a pseudo teletype. Although this project started out largely as a toy, large numbers of users across the country are starting to express serious interest.

#### 8) Rand Software

The Rand editor has undergone some improvement to allow it to execute external programs for text processing. So far interfaces to "nroff", for justification, and to a program called

"rpt", for global replacement, have been completed.

A "message system" is undergoing testing which manages very high-volume mailboxes in a manner efficient to the user. It treats each component of each message (from, to, cc, etc.) separately, and allows the user to compose messages with arbitrary components. It manages permanent mailboxes so that old messages may be forwarded, deleted, undeleted, edited (!), etc. Each component of a message may be separately edited.

In order to allow as much as possible in the way of customization of commands, a tool has been added to handle the plethora of extra flags such customizations require for commands. Called "monitoring", this is an extra feature residing between the user and his shell, which expands commands for which it has a profile to include sets of flags, arguments and pre- and post-commands specified by the user in his permanent "profile" for that command. String substitution with arbitrary-length strings is also available.

#### 9) Odds and Ends

Being a collection of raw random data, odd thoughts, and conversations that don't fit in anywhere else. There was a lot of this toward the end of the meeting.

A Personal Computer Network, being set up around the Bay Area by computer hobbyists (including some ARPA people), is going to have a first protocol ready to try out in a month or so. Initially connections will be made via land-line, going over to amateur radio when the FCC pulls its head out of the ground and allows ASCII on the ham bands.

Jeff Rotman at UC-Berkeley, when he was at a school on the East Coast (unfortunately I don't remember which one --Ed.) made some changes to UNIX to give it a real-time capability. Currently real-time processes live in Supervisor space and field interrupts directly from the device registers, completely outside the rest of UNIX. They have hooks into the system so that normal UNIX processes may read the data that the Supervisor process has fielded. The normal UNIX processes are locked in core by a separate flag from SLOCK. It is being contemplated that the Supervisor process might be done away with entirely, depending on what sort of speed is required.

Alan Stoughton at UCLA has a driver for Dykstra semaphores, so that user programs can do real P and V type things. The Ingres group at UCB has developed a slightly different type of semaphore for use in their DBMS work.

Ed Gould at UC Berkeley has a new shell that looks like a language. It has C-style syntax.

Something has to be done about UNIX power failures since it probably shouldn't try to do an update with no power. A branch loop might be appropriate. Or, to be fancier, have the clock

wait 20 seconds and then do an automatic reboot. This eliminates the problems of power fluctuations often seen in real situations when the power has gone funny.

A related item is to have a check made such that if "init" should die, the system would automatically reboot.

Some people would like a blocking version of the "sync" system call, so that in situations where there is a fairly huge number of in-core buffers which might have to be written out, you can resume processing with the assurance that the disk is fully up-to-date.

Instrumentation - code left in m45.s by Ken Thompson may be used to profile the system. He used the o-clock at high speed and reserved an area of core as large as the text size of the system. The pc was used as an index into this monster array and the corresponding cell incremented at every tick. He ran this for a month and got some very meaningful statistics on system utilization. In particular he found that a large amount of time was spent searching the process table.

Data base systems - INGRES is available from Bob Epstein at UC Berkeley, Electronics Research Lab. New release is V6. Boston Children's Museum also has a very fast dictionary-type database system available for money. The RAIN Relational Algebra Interpreter is being developed by Prof. Shi-Kuo Chang, Dept. of Information Engineering, U. of Illinois @ Chicago Circle, PO Box 4348, Chicago Ill. 60680 in a medical database environment, though it is general-purpose.

Ari Ollikainen is drafting a letter from the entire West Coast Unix Users' Group to DEC field service protesting the degradation in performance DEC has shown of late. At least two installations have told DEC that if certain service personnel are sent to their sites ever again, payment will be withheld.

System call #64 is being reserved for installation-dependent indirect calls, and system call #63 for Users' Group standard calls. No drop-in code to support these indirect calls has been developed yet (good luck, gang!).

#### 10) Coordination of changes

There was a great deal of discussion about the coordination of system changes being made all over the country. Several observations emerged.

The UUG distributions are very hard to use. There should be a brief (for each piece of improved software) of why it is different, how it is different, and what non-stock system features it relies upon. In the distributions as they now exist, you get a list of what files are on the tape. Some have reasonable documentation, some don't. There is no way to tell what software you want without taking each file off of tape and diffing it with the

analogous source that you have.

A form for describing a changed piece of software should be formalized, and circulated. Then, the filled out forms, should be put online somewhere, so that a person interested in some changes to program X can find out who has already done what kinds of things to program X. Jim Lieb of ISI has volunteered some of his time, disk and cycles to try to make this happen.

Mark Kampe suggested the use of standard, in-code, prose forms for program documentation. Such forms are in use at UCLA and Univ of Ill for all system changes and new pieces of software. The objection was raised that such comments in the code can substantially increase the size of source files on disk. There are clearly two schools of thought on in-code documentation. Sample forms are included in an appendix with comments on their use.

It was suggested, that when a program is altered, that a diff be sent to the central clearing house as a useful piece of documentation. It was countered that diff often produces unreadable output. Jim Lieb mentioned that a nicer differential file comparison program is now available.

John Bass mentioned that he has, for some time, been using conditional compilation to out in his changes. This allows him to easily turn off features he doesn't want, at no object code cost. It has the nice side effect of clearly bracketing off the changed code.

Jim Lieb is going to attempt to start a database of system changes. This is an attempt to increase the ambient level of understanding about what changes are available and who has been working on what. Also, it can serve as a shopping list for new Unix sites who are pondering what system features they want to get, and have no idea what is available.

#### Suggested forms for In-Code documentation

During a discussion on distribution packages, the use of a standard form for incode documentation was mentioned. This form was devised at the University of Illinois Center for Advanced Computation and has proved very popular with everyone who has seen it.

This header appears at the beginning of every major routine in the Illinois MCP, and of every major piece of Unix software written by our group at UCLA. At first, we thought it would be a pain to fill out this form for each routine. Our thoughts on that subject have changed. We now fill out the form before coding the routine. If you don't know enough to fill out the form, you clearly aren't ready to write code. If you have the form filled out, the code goes really quickly.

We have found that this religious use of this form makes it easier for a new person to read and understand the code, easier for the programmer to recomprehend the code, and easier for the programmer to hand the program off to another programmer. In addition, collecting together all of these forms, and adding a small amount of general prose, results in a package of structural documentation superior to almost anything else we have seen.

It is hoped that other sites, who are concerned about the above issues, will examine this form and decide that they want to use one like it. What is important is not this form, or any particular form, but rather that some effort is made to raise the ambient level of documentation.

```
/*L name:
    <fill in the name of the routine>

function:
    <describe the function performed by this routine>

algorithm:
    <describe, in general, the manipulations which are
    performed by this routine. This should serve as
    a map through the code>

parameters:
    <list the names, types and significances of the parameters>

returns:
    <list the classes of returned values and their significances>

globals:
    <list all globals referenced or modified by this routine>
    it is particularly cool to designate which ones are modified>

calls:
    <list all routines called by this routine>

called by:
    <list all routines which call this routine
    this is the hardest part to keep updated>

history:
    <starting with the original writer of the routine
    detail each time the routine has been changed, by
    whom and why.
    the contents of this field are clearly the least
    well defined

/*L/
```

This form was devised at UCLA as a counterpart to the above form. We use it at the beginning of each file of code. It goes at the beginning of each module.

```
/*-L
module name:
  <name of the file goes here>

function:
  <describe what this module has to do with the overall system>

globals contained:
  <enumerate the globals defined in this module and their purposes>

routines contained:
  <enumerate the routines defined in this module and their purposes>

modules referenced:
  <enumerate the modules that are included in the compilation of
  this module and explain their significances>

modules referencing:
  <enumerate the modules which, in some fashion, depend on this
  module. As above, this field is ill defined and hard to maintain>

compile time parameters:
  <enumerate the compile time parameters and their effects>

history:
  <it is a matter of personal taste, what things are module history
  and what things are routine history>

-L */
```

We keep both of these forms in generally readable files, and just read them into the editor when we are typing in code. Our experience (direct and from people we have given our code to) is that the use of these files adds considerably to the ease of writing, maintaining and studying code.

We have received (from a few sites) the objection that including all this prose in the code dramatically increases the size of the source code. We find that to be a small price to pay, but can understand that not all groups would agree.

These forms have been shown in abstract. We (and most other people) fell in love with them, not when they were described, but rather when we were trying to understand a piece of code which included them. Something which can change, overnight, the coding styles of thirty people must be fairly impressive.

## Appendix II Wants and Haves List

### Wants:

DA11 B driver (RAND)  
Improved RPD4 driver (RAND)  
Improved TU10 driver (RAND)

Floppy driver (NOSC)  
A way of USEFULLY using mag tape  
for read/only file systems  
(NOSC)

BMC11 driver (OSI)  
256K Mode Lisp (OSI)  
11/04 Unix (OSI)

DQ driver (BNR)

troff -a handling (SRI)

Unix Systems and applications types  
(SRI, BNR)

### Haves:

Tenex Style echoing in improved tty driver  
(NOSC, UCLA)

VBN Driver for the ILL MCP

Help system for the Unix Programmers Manual  
(UCSB)

Multifile TU10 Support  
(UCSF, UCLA)

Genesco graphics support  
(RAND)

Semaphore device driver  
(UCB, UCLA)

Appendix III  
Bug report and survey forms

Jim Lieb is going to attempt to act as a more formal clearing house for bugs and fixes and information about different peoples versions of the system. He has proposed two forms. One for bug reports and one for description of system changes. He is going to attempt to build databases of the collected information and make them generally available.

The first form is for bug reports. It is hoped that the existence of a formal, centralized and organized system for collecting bugs and fixes will result in the more rapid propagation of fixes through the community and reduce duplication of effort. Jim will, in the near future, distribute information about how his bug database can be interrogated and how reports of that information can be obtained.

The second form is for a survey of system changes and versions. The results of this survey will go into a database and will be made available to people interested in enhancements to unix. It is hoped that placing this information online, and attempting to maintain a database will make it a valuable resource to the entire community. The mechanics of the distribution and accessing of this information will be described soon.

completed forms should be mailed to:

Jim Lieb  
USC/Information Sciences Institute  
4676 Admiralty Wy  
Marina Del Rey, Ca 90291

or LIEB@ISI@



The  
University of Oklahoma

202 West Boyd, Room 219 Norman, Oklahoma 73019

School of Electrical Engineering  
and Computing Sciences

October 5, 1977

Professor Melvin Perentz  
c/o CUNY/UCC  
555 West 57 Street  
New York, NY 10019

Dear Professor Perentz:

I am probably not the only person who has more than once reread all the old issues of "login" to see if anyone has a driver for a particular peripheral. For my own gratification, and those in a similar position, I am offering to make a first cut at a directory of alien and/or modified device drivers. If people who have drivers for non-stock devices or have heavily modified a driver for some particular reason would send me a short note describing their creations, I will collect all the loose ends and then forward a condensed listing to you for possible publication. The note should describe the device, the reasons behind modifications, and its availability (author will distribute; author won't distribute but will give hints to others; etc.).

I will try to get the list to you on about December 1 so people will have a chance to respond.

Thanks for the help.

Sincerely,

Michael D. O'Dell  
Research Assistant  
Computer Science Group  
University of Oklahoma  
202 West Boyd, Room 219  
Norman, OK 73019

MO/ba

\*\*\*\*\*UNIX SITE SURVEY\*\*\*\*\*

=====

General Information

Site Name:  
Site (mailing/Arpanet)  
Address:

System Manager/Administrator:  
Software support person:

=====

System information

Cpu: PDP 11/                      Unix system version (Bell Labs):  
Current version (virgin Bell Labs or enhanced by another site):  
Date you received this version:  
Enhancements made at your site:                      (describe interesting and useful things  
you changed/added to EXISTING software)

Describe NEW software you have developed at your site and are willing  
to release to the community:

Subquestions:

Is it documented?

Will you support it?

Will you provide sources?

Will you provide consultation services?

    If so how much and how can you be contacted?

Do you plan enhancements and new releases?

Is there any licensing (\$\$) involved?

    If so, please describe:

Describe your software wants/needs:                      (be as unrealistic as you like. Someone  
just may have it.)

Are you willing/able to provide system resources or personal resources to  
the Unix community?

    If so, please describe conditions and arrangements:

Everything else you would like to say and we forgot to ask:

# UNIX BUG REPORT

Date: Cpu: PDP 11/ Unix version:

Reported by (name):

Site name:

Address:

(Street)

(City and state)

(Zip code)

(Telephone)

(ext.)

Please reply: Immediately ( )

In "Login" ( )

By Mail ( )

Don't reply ( )

Type of report:

Program error ( )

Suggestion ( )

Complaint ( )

Documentation error ( )

Request for information ( )

For your information ( )

Module or document affected:

Unix kernel ( )

Command (Bell) ( )

Document (Bell) ( )

Library (Bell) ( )

Program (Bell) ( )

Unix shell ( )

state which/whose version:

Command (Unix group) ( )

Document (Unix group) ( )

Library (Unix group) ( )

Program (Unix group) ( )

Module or document name:

If it is a programming error is it reproducible? (yes/no):

Do you have a solution/correction? (yes/no):

Description of problem. Include source or diff listing if you can:

Correction or fix suggested:

Received by:

Repaired by:

Answered by:

date:

date:

date: